FlowCoder

GLOSSARY

용어 사전

처음 만나는 단어들

낯선 단어가 나오면 여기서 찾아보세요.
전부 외울 필요는 없습니다.

비개발자를 위한 AI 실무 용어 219개 · 복사해 쓰는 프롬프트 15개

검색해서 찾으려면 - https://flowcoder-ai-glossary.vercel.app

이 사전을 쓰는 법

- 실습은 화면을 그대로 따라 하면 됩니다. 이 사전은 "아까 그 단어가 뭐였지?" 할 때 펼치는 용도입니다.
- 컴퓨터를 잘 몰라도 읽히도록 적었습니다. 어원이나 기술적 배경은 일부러 쓰지 않았습니다.
- 오늘 강의에 나오지 않는 단어도 들어 있습니다. 안 나오면 그냥 넘기시면 됩니다.
- 뒤쪽 찾아보기에서 단어를 찾으면 어느 묶음에 있는지 바로 나옵니다.

CONTENTS

A-1 화면과 도구 5
A-2 파일과 폴더 5
A-3 AI 기본 5
B-1 에이전트와 자동화 6
B-2 도구와 규칙 파일 7
B-3 계정과 연결 7
B-4 시트·데이터 자동화 8
C-1 기획 프레임 8
C-2 보안 9
C-3 평가·검증 9
C-4 결과물·자산 10
D-1 버전 관리 (Git·GitHub) 11
D-2 배포와 서버 11
D-3 개발자와 일할 때 나오는 말 12
D-4 화면에 뜨는 것 읽기 13
E-1 키보드 단축키 13
E-2 입력 앞에 붙이는 기호 14
E-3 자주 쓰는 슬래시 명령 14
E-4 손에 익힐 것과 함정 14
F-1 확인할 때 15
F-2 막혔을 때 15
F-3 시킬 때 16
F-4 고쳐 달라고 할 때 17
G-1 도구가 안 될 때 18
G-2 결과가 미덥지 않을 때 18
G-3 파일·저장에서 막힐 때 19
H-1 클라우드와 서버 19
H-2 회사에서 듣는 보안 용어 20
H-3 AI가 짜 준 코드 읽기 20
H-4 이름만 알아두면 되는 것들 21
H-5 공장·설비에서 듣는 말 21

오늘 이것만 먼저

219개를 다 볼 필요는 없습니다. 어느 강의에서나 그날 안에 쓰이는 12개만 먼저 훑으세요. 나머지는 필요할 때 찾아보면 됩니다.

프롬프트 (지시문)	AI에게 보내는 요청문입니다. 채팅창에 쓰는 그 문장입니다.
토큰	AI가 글을 세는 단위로, 단어보다 잘게 쪼갠 조각입니다. 사용량과 비용이 이 단위로 계산됩니다. 무제한이 아닙니다.
컨텍스트 윈도우	AI가 한 번에 기억하며 볼 수 있는 작업 책상의 크기입니다. 넘치면 앞 내용을 흘립니다.
환각 (Hallucination)	AI가 없는 사실을 그럴듯하게 지어내는 현상입니다. "자료에 있는 내용만 근거로 답해줘"를 붙이면 눈에 띄게 줄어듭니다. 다만 없어지지는 않으니 숫자와 인용은 원본과 대조하세요. → G-2
RAG	관련 문서를 먼저 찾아 읽고 그 내용을 근거로 답하는 방식입니다. 시험으로 치면 오픈북입니다. 다만 찾아온 문서가 엉뚱하면 답도 엉뚱해집니다.
Agent (에이전트) vs Chatbot	Chatbot은 묻고 답하는 대화가 중심입니다. 여러 번 주고받을 수는 있지만 실제 실행(파일 열기·저장·발송)은 사람이 합니다. Agent는 목표만 주면 여러 단계를 스스로 밟아 결과물 파일까지 만듭니다. 갈림길은 대화 횟수가 아니라 스스로 실행하는가입니다.
권한 (승인)	"이 프로그램이 내 문서를 봐도 되나요?"라는 물음에 허락하는 것입니다. 무엇에 대한 허락인지 확인하고 누르는 습관이 중요합니다.
MCP	AI와 바깥 도구를 잇는 표준 연결 규격입니다. 연결 다리 또는 콘센트 규격이라고 보면 됩니다. 규격만 맞으면 어떤 도구든 꽂습니다.
검산	"이 값을 넣으면 이 값이 나와야 한다"는 확인용 예시 한 줄입니다. 결과가 맞는지 재는 자입니다. 여러 건을 종류별로 모으면 Test Case(C-3)가 됩니다.
기준 문서	AI의 결과물이 "됐다"인지 판정할 때 대조하는 문서입니다. 판정 기준을 AI 바깥에 둡니다.
가명 / 마스킹	민감한 이름·숫자를 `A사` · `○○`처럼 가리는 것입니다. 민감 데이터는 마스킹한 뒤에만 다룹니다.
올려도 되나 (3초 판단)	사람 이름·연락처·주민번호·계좌, 아직 공개되지 않은 실적·단가·계약 내용은 그대로 올리지 않습니다. 이름은 `A사` · `김○○`로 바꾸고, 형식만 확인할 때는 더미 데이터로 대신합니다. 애매하면 회사 규정과 담당 부서를 먼저 확인합니다.

A. 환경·기초

A-1 화면과 도구

터미널 (PowerShell·CMD·터미널)	마우스 클릭 대신 글자로 명령을 거는 창입니다. 검은 바탕에 글자만 나와 무섭게 생겼지만, 실제로 쓰는 명령은 몇 개뿐입니다. Windows에서는 PowerShell·CMD, Mac에서는 터미널이 이에 해당합니다.
브라우저	인터넷을 보는 프로그램(크롬·엣지 등)입니다. 웹으로 쓰는 AI 도구는 전부 이 위에서 돌아갑니다.
편집기 (VS Code 등)	파일 내용을 눈으로 보며 고치는 프로그램입니다. 메모장의 전문가판이라고 보면 됩니다. 없어도 실습은 됩니다.

A-2 파일과 폴더

파일 / 폴더	파일은 내용물 하나(문서·표·사진), 폴더는 파일을 담는 상자입니다.
작업 폴더 · 경로	작업 폴더는 오늘 다룰 파일을 모아둔 폴더입니다. AI 도구는 보통 실행한 폴더 안만 봅니다. 경로는 파일이 어디 있는지 나타내는 주소입니다.
확장자	파일 이름 끝에 붙는 `.md` `.csv` 같은 꼬리표입니다. 파일의 종류 표시로, 컴퓨터는 이걸 보고 어떤 프로그램으로 열지 정합니다.
.md (마크다운)	글자만 있는 메모 파일입니다. `#`·`-` 같은 기호로 제목과 목록을 표시하고 메모장으로도 열립니다. 규칙서·기획서가 대부분 이 형식입니다.
.csv · .xlsx	`.csv`는 쉼표로 칸을 나눈 표 파일, `.xlsx`는 엑셀 문서 파일입니다. 둘 다 엑셀·스프레드시트로 열립니다.
.html	브라우저로 여는 웹페이지 파일입니다. 계산기·대시보드 같은 산출물이 이 형식으로 나옵니다.
.zip	여러 파일을 하나로 묶은 압축 파일입니다. 마우스 오른쪽 → 압축 풀기로 원래대로 꺼냅니다.
덮어쓰기 vs 이어붙이기 (append)	덮어쓰기는 기존 내용을 지우고 새로 쓰는 것이라 위험합니다. 이어붙이기(append)는 기존 내용을 두고 아래에 한 줄 더하는 것이라 안전합니다. 기록이 쌓이는 자동화의 기본 규칙은 "항상 이어붙이기"입니다.

A-3 AI 기본

생성형 AI · LLM	생성형 AI는 글·그림·코드를 만들어내는 AI입니다. LLM은 그중 언어를 다루는 대형 언어 모델(Claude·GPT류)로, "언어를 다루는 AI"로 기억하면 충분합니다.
모델 · 기반 모델 (Foundation Model)	AI의 두뇌에 해당하는 프로그램입니다. "모델 학습에 쓰지 않는다"는 약속은, 사용자가 넣은 데이터로 이 두뇌를 다시 가르치지 않는다는 뜻입니다.

프롬프트 (지시문)	AI에게 보내는 요청문입니다. 채팅창에 쓰는 그 문장입니다.
프롬프트 5원칙	지시문에 채워 넣을 다섯 칸입니다 - 역할·맥락·제약·형식·예시. 빠뜨리기 쉬운 항목을 챙기게 하는 체크리스트이고, 그중 제약 칸이 결과를 가장 크게 바꿉니다. 빈칸 틀은 F-3에 있습니다.
Few-shot (예시 제시)	정답 예시를 한두 건 먼저 보여줘 결과의 형식을 고정하는 방법입니다. 말로 설명하는 것보다 예시 하나가 빠릅니다.
XML 태그	`<report>` 처럼 꺾쇠로 감싼 칸 이름표입니다. AI가 매번 같은 칸에 같은 내용을 넣게 만드는 장치입니다.
토큰	AI가 글을 세는 단위로, 단어보다 잘게 쪼갠 조각입니다. 사용량과 비용이 이 단위로 계산됩니다. 무제한이 아닙니다.
컨텍스트 윈도우	AI가 한 번에 기억하며 볼 수 있는 작업 책상의 크기입니다. 넘치면 앞 내용을 흘립니다.
환각 (Hallucination)	AI가 없는 사실을 그럴듯하게 지어내는 현상입니다. "자료에 있는 내용만 근거로 답해줘"를 붙이면 눈에 띄게 줄어듭니다. 다만 없어지지는 않으니 숫자와 인용은 원본과 대조하세요. → G-2
멀티모달	글자만이 아니라 이미지·음성·표까지 읽고 만드는 능력입니다.
RAG	관련 문서를 먼저 찾아 읽고 그 내용을 근거로 답하는 방식입니다. 시험으로 치면 오픈북입니다. 다만 찾아온 문서가 엉뚱하면 답도 엉뚱해집니다.

B. 에이전트·도구

B-1 에이전트와 자동화

Agent (에이전트) vs Chatbot	Chatbot은 묻고 답하는 대화가 중심입니다. 여러 번 주고받을 수는 있지만 실제 실행(파일 열기·저장·발송)은 사람이 합니다. Agent는 목표만 주면 여러 단계를 스스로 밟아 결과물 파일까지 만듭니다. 갈림길은 대화 횟수가 아니라 스스로 실행하는가입니다.
메인 에이전트 / 서브에이전트	메인은 작업 공간을 운영하는 팀장, 서브는 그 아래에서 한 가지 일을 맡는 직원입니다. 여러 명을 동시에 부릴 수 있습니다.
병렬 / 순차 (파이프라인)	병렬은 서로 상관없는 일을 여러 직원이 동시에 하는 것, 순차는 앞사람 결과가 나와야 다음이 시작되는 것입니다. 순차를 파이프라인 또는 협업이라고 부르기도 합니다.
Tool Use (도구 호출) · Function Calling	AI가 파일 읽기·계산·검색 같은 도구를 스스로 골라 쓰는 것입니다. Function Calling(함수 호출)에서 넓어진 개념으로, Tool Use가 함수까지 품는 더 큰 말입니다.
Sub-goal · 멀티스텝	Sub-goal은 큰 목표를 이루려고 쪼갠 작은 단계(보통 3~5개)입니다. 멀티스텝은 한 번에 끝나지 않고 여러 단계와 갈림길을 거치는 작업을 말합니다.

바이브 코딩	코딩 없이 자연어로 설명하면 AI가 코드를 대신 짜 주는 방식입니다. 내가 못 하는 것을 말로 위임하는 감각입니다.
E2E (End-to-End) 자동화	입력 → 처리 → 출력을 하나의 흐름으로 잇는 자동화입니다. 조각이 아니라 처음부터 끝까지입니다.
3단 스케치	Trigger → Action → Output. 자동화 후보를 "무엇이 들어오면 · 무엇을 해서 · 무엇이 나온다" 세 칸으로 그리는 틀입니다.
로컬 스크립트	AI가 만들어 준, AI 없이 내 컴퓨터에서 혼자 도는 자동화 프로그램입니다. 판단이 필요 없는 정형 계산을 맡깁니다.

B-2 도구와 규칙 파일

Cowork Project	지시·파일·메모리를 담아 두는 저장된 나만의 도우미입니다.
Live Artifacts	다시 열고 고칠 수 있는 실행형 결과 화면(표·미리보기·대시보드)입니다.
Claude Code CLI	터미널에서 쓰는 Claude입니다. CLI는 "명령줄 도구"라는 뜻의 영어 약자로, 터미널에서 글자로 부리는 도구라고 기억하면 충분합니다.
슬래시 명령	`/`로 시작하는 짧은 명령(예: `/clear`, `/compact`)입니다. 대화 대신 버튼처럼 누르는 지름길이고, 직접 만든 명령도 같은 자리에 붙습니다.
Skill · SKILL.md	반복 작업 절차를 파일로 굳혀 이름으로 다시 부르는 것입니다. 직접 만들 수 있고, 내일도 동료도 씁니다.
플러그인 · skill-creator	플러그인은 도구에 끼우는 부가 기능입니다. skill-creator는 Skill을 잘 만들게 도와주는 공식 플러그인입니다.
CLAUDE.md (규칙서)	작업 폴더에 두면 시작할 때 자동으로 읽히는 규칙 파일입니다. "늘 한국어로"·"덮어쓰기 금지" 같은 약속을 담습니다. 그때그때 치는 프롬프트와 달리, 저장해 두면 매번 적용됩니다.
하네스 (판)	에이전트가 일하는 판입니다. 작업 폴더·규칙 문서·기준 문서·재사용 절차로 짭니다. 에이전트는 모델과 하네스의 합입니다.

B-3 계정과 연결

계정 / 워크스페이스	계정은 내 로그인 신분증, 워크스페이스는 회사 단위의 작업 공간입니다.
로그인 / 인증	내가 나임을 증명하는 절차입니다. 화면 안내를 따라가면 됩니다.
권한 (승인)	"이 프로그램이 내 문서를 봐도 되나요?"라는 물음에 허락하는 것입니다. 무엇에 대한 허락인지 확인하고 누르는 습관이 중요합니다.
API	프로그램끼리 데이터를 주고받는 창구입니다. 사람 대신 프로그램이 드나드는 출입구입니다.

MCP	AI와 바깥 도구를 잇는 표준 연결 규격입니다. 연결 다리 또는 콘센트 규격이라고 보면 됩니다. 규격만 맞으면 어떤 도구든 꽂습니다.
커넥터	챗 화면에서 외부 서비스(메일·캘린더·드라이브 등)를 클릭 몇 번으로 잇는 연결 방식입니다.
브라우저 확장	AI가 내 브라우저를 직접 읽고 움직이게 하는 연결입니다. 로그인 상태를 공유하므로 실습은 공개된 웹에서만 합니다.

B-4 시트·데이터 자동화

Google Sheets	구글의 온라인 엑셀입니다. 주소창에 `sheets.new`를 치면 새 시트가 열립니다.
시트 ID	시트 주소(URL) 가운데 있는 고유 번호입니다. 코드가 "어느 시트인지" 찾을 때 씁니다.
Apps Script	구글 문서·시트에 내장된 자동화 도구입니다. 별도 설치 없이 시트 상단의 확장 프로그램 메뉴에서 엽니다.
스크립트 / 함수	스크립트는 자동으로 실행되는 코드 묶음, 함수는 그 안에서 한 가지 일을 맡는 단위입니다.
트리거 (시간 트리거)	"매일 오후 6시" 같은 조건이 되면 자동으로 실행해 주는 예약 장치입니다. 내가 없어도 돕니다. B-1 3단 스케치의 Trigger는 "무엇이 들어오면"이라는 칸 이름이고, 여기 트리거는 그 칸을 시간으로 채운 실물 장치입니다.

C. 프레임·보안·평가

C-1 기획 프레임

페인포인트	반복돼서 시간을 잡아먹는 내 업무의 아픈 지점입니다. 자동화 후보를 고르는 출발점입니다.
5 Whys	"왜?"를 반복해 물어(보통 세 번에서 다섯 번) 눈에 보이는 불편(증상) 밑의 진짜 원인까지 파는 방법입니다.
SCAMPER	일곱 가지 질문(대체·결합·응용·변형·다른 용도·제거·반전)으로 업무를 비틀어 아이디어를 넓히는 도구입니다.
ERRC	넓힌 아이디어를 제거·감소·향상·창출 네 칸으로 분류해, 실행할 것만 골라 모으는 도구입니다.
문제 정의 카드	벼린 페인포인트를 다섯 줄(문제·진짜 원인·골라낸 방향·성공 기준·사람 확인)로 압축한 인수인계 카드입니다.
Job to be Done	이 일이 진짜 이뤄 주려는 목적입니다. 겉으로 하는 작업 말고, 그 작업이 대신해 주는 일을 봅니다.
As-is / To-be	지금 상태와, 바뀐 뒤의 상태입니다. 둘을 나란히 적으면 무엇을 바꾸려는지가 분명해집니다.

PRD · PRD 메모	PRD는 무엇을·왜·어떤 결과물로 만들지 적은 요구사항 문서(작업표준서)입니다. PRD 메모는 그 세 줄 축약판입니다.
SPEC · PLAN	기획 3단의 뒷 두 단입니다. SPEC은 어떤 모양으로, PLAN은 어떤 순서로. PRD와 합쳐 기획 3단이 됩니다.
검산	"이 값을 넣으면 이 값이 나와야 한다"는 확인용 예시 한 줄입니다. 결과가 맞는지 재는 자입니다. 여러 건을 종류별로 모으면 Test Case(C-3)가 됩니다.
기준 문서	AI의 결과물이 "됐다"인지 판정할 때 대조하는 문서입니다. 판정 기준을 AI 바깥에 둡니다.
SMART 목표	막연한 바람을 구체·측정 가능·달성 가능·관련성·기한이 있는 목표 문장으로 바꾸는 틀입니다.
성공 지표	목표 달성을 숫자로 확인하는 잣대입니다. 예를 들면 처리 시간 40분 → 15분.
Quick Win / Long Bet	임팩트와 난이도로 갈랐을 때 "지금 바로 할 것"과 "길게 볼 것"의 구분입니다.
파일럿 · Phase	파일럿은 먼저 나 혼자 시험 삼아 써 보는 1단계입니다. Phase는 단계라는 뜻이고, 이 교육에서는 Phase 1을 골격 만들기, Phase 2를 검증으로 씁니다.
PBL · 확산 계획서	PBL은 실제 과제를 프로젝트로 직접 수행하며 배우는 방식입니다. 확산 계획서는 만든 것을 팀·부서로 넓히는 한 장짜리 설계도입니다.

C-2 보안

LV1~LV4 (데이터 4계층)	데이터 보안 4계층입니다. LV1 공개 · LV2 내부 · LV3 민감 · LV4 기밀. 위로 갈수록 AI에게 덜 보여줍니다.
가명 / 마스킹	민감한 이름·숫자를 `A사` · `○○`처럼 가리는 것입니다. 민감 데이터는 마스킹한 뒤에만 다룹니다.
더미 데이터	진짜처럼 생긴 가짜 연습용 데이터입니다. 기밀 데이터는 더미로만 흐름을 검증합니다.
벤더 정책 · ZDR	AI 회사가 약속하는 데이터 취급 조건입니다. 학습에 쓰지 않음, 보존 기간, ZDR(아예 저장하지 않음) 등이 여기에 들어갑니다.
올려도 되나 (3초 판단)	사람 이름·연락처·주민번호·계좌, 아직 공개되지 않은 실적·단가·계약 내용은 그대로 올리지 않습니다. 이름은 `A사` · `김○○`로 바꾸고, 형식만 확인할 때는 더미 데이터로 대신합니다. 애매하면 회사 규정과 담당 부서를 먼저 확인합니다.
최소 권한	필요한 만큼만 허락하는 원칙입니다. 읽기만 필요하면 쓰기 권한은 주지 않습니다.

C-3 평가·검증

AI Validation (검증)	만든 AI를 믿고 실무에 써도 되는지 확인하는 일입니다. 기준을 직접 만드는 것부터가 검증입니다.

평가 대상 3층위	문제가 생기는 곳을 Model(두뇌)·System(내가 짠 판)·Product(쓰임새) 세 층으로 나눠 보는 틀입니다.
Test Case	AI를 시험해 볼 입력 한 건입니다. 정상·결측·이상치처럼 종류를 나눠 설계합니다.
Rubric (채점 기준표)	평가 기준을 표로 미리 정해 점수를 매기는 방식입니다. 채점 기준표에 해당합니다.
LLM-as-a-Judge	그 채점을 언어 AI에게 대신 시키는 것입니다. 기준표를 든 심사위원입니다.
Human Eval	사람이 직접 산출물을 보고 품질을 판정하는 방법입니다. 시식단에 해당합니다.
Benchmark	남들이 만들어 공개한 표준 문제 세트입니다. 모델이나 버전을 같은 잣대로 견줄 때 씁니다. 모의고사에 해당합니다.
Golden Set	우리 업무 데이터로 우리가 직접 정답을 합의해 만든 문제 세트입니다. 바깥 모의고사(Benchmark)와 달리 우리 기준이라, 우리 결과물 판정에는 이쪽이 맞습니다.
Regression (회귀 확인)	고친 뒤에도 예전 문제를 다시 풀려 망가지지 않았는지 확인하는 것입니다. 재시험에 해당합니다.
Inter-rater Agreement	두 사람이 따로 채점해도 점수가 비슷한 정도입니다. 채점자끼리 손발이 맞는가를 봅니다.
Guardrail	AI가 틀린 숫자나 지어낸 값으로 벗어나지 않게 막는 상시 지시입니다. 규칙 파일(CLAUDE.md, B-2)에 적어 두면 매번 자동으로 걸립니다.
개선 로그	무엇을 고쳤고 점수가 어떻게 변했는지 남기는 기록입니다. 고칠수록 나아진다는 증거가 됩니다.

C-4 결과물·자산

대시보드	흩어진 숫자를 한 화면에 모아 보여주는 상황판입니다. 자동차 계기판이 원조입니다.
KPI	업무 상태를 보여주는 핵심 숫자입니다. 월 매출, 문의 건수, 반품률 같은 것입니다.
목표선	차트에 그어두는 기준선입니다. 목표를 넘었는지 못 미쳤는지 한눈에 보입니다.
템플릿	형식이 미리 잡혀 있어 값만 바꿔 쓰는 틀입니다.
자산 / 메타데이터	자산은 재사용할 도구(Skill·프롬프트·템플릿)입니다. 메타데이터는 자산에 붙이는 꼬리표(이름·용도·입출력·담당)로, 도서관 청구기호 같은 것입니다.

D. 개발자 말 알아듣기

D-1 버전 관리 (Git·GitHub)

Git / GitHub	Git은 파일이 언제 어떻게 바뀌었는지 이력을 남기는 도구입니다. GitHub은 그 이력을 인터넷에 올려 여럿이 함께 쓰는 서비스입니다. 둘은 다른 것입니다.
저장소 (repo)	프로젝트 하나가 통째로 들어가는 폴더와 그 변경 이력입니다. 리포·레포라고 줄여 부릅니다.
커밋 (commit)	지금 상태에 이름을 붙여 저장하는 것입니다. 게임의 세이브에 해당하고, 언제든 그 지점으로 돌아갈 수 있습니다.
브랜치 (branch)	본선을 건드리지 않고 옆에서 따로 작업하는 갈래입니다. 망쳐도 본선은 멀쩡합니다.
main (본선)	기준이 되는 갈래입니다. 실제로 서비스되는 것은 보통 이 갈래입니다. 지금도 master라는 이름을 쓰는 저장소가 있습니다.
머지 (merge)	따로 작업한 갈래를 본선에 합치는 것입니다.
충돌 (conflict)	같은 줄을 두 사람이 다르게 고쳐서 자동으로 합쳐지지 않는 상태입니다. 사람이 어느 쪽을 남길지 골라야 풀립니다.
PR (Pull Request)	"이 갈래를 본선에 합쳐도 될까요"라고 올리는 요청입니다. 합치기 전에 남이 읽고 의견을 다는 자리이기도 합니다.
클론 / 푸시 / 풀	클론은 저장소를 내 컴퓨터로 통째로 내려받기, 푸시는 내 변경을 올리기, 풀은 남의 변경을 받아오기입니다.
되돌리기 (revert)	잘못 넣은 커밋의 내용을 무효화하는 새 커밋을 하나 더 얹는 것입니다. 과거 이력을 지우지 않아, 되돌렸다는 사실까지 기록에 남습니다. 배포한 것을 통째로 되돌리는 것은 롤백(D-2)입니다.
.gitignore	저장소에 올리지 않을 파일 목록입니다. 비밀번호가 든 파일이나 임시 파일을 여기에 적어 막습니다. 환경변수 파일(`.env`, D-2)이 대표적인 대상입니다.
README	저장소 첫 화면에 뜨는 설명서입니다. 이게 뭐고 어떻게 쓰는지를 적습니다.

D-2 배포와 서버

로컬 / 서버	로컬은 내 컴퓨터, 서버는 남들이 인터넷으로 접속하는 컴퓨터입니다. "로컬에서는 되는데요"는 내 컴퓨터에서만 된다는 뜻입니다.
빌드 (build)	사람이 쓴 코드를 실제로 돌아가는 형태로 굽는 과정입니다. 빌드가 깨졌다는 말은 이 단계에서 실패했다는 뜻입니다.
배포 (deploy)	만든 것을 남이 쓸 수 있는 자리에 올리는 일입니다.

호스팅	그 자리를 빌려주는 서비스입니다. 웹사이트를 얹어 두는 땅이라고 보면 됩니다.
도메인 / URL	도메인은 `flow-coder.com` 같은 주소 이름, URL은 특정 화면까지 가리키는 전체 주소입니다.
스테이징 / 프로덕션(운영)	스테이징은 손님을 받기 전 연습 무대, 프로덕션은 실제 손님이 쓰는 무대입니다. 사고는 프로덕션에서 납니다.
프론트엔드 / 백엔드	프론트엔드는 눈에 보이는 화면, 백엔드는 뒤에서 도는 계산과 저장입니다. 식당으로 치면 홀과 주방입니다.
데이터베이스 (DB)	데이터를 담아 두고 찾아 쓰는 창고입니다. 아주 큰 엑셀이라고 생각해도 처음에는 무리가 없습니다.
환경변수 (.env)	코드 바깥에 따로 두는 설정값입니다. 비밀번호·API 키를 코드에 박지 않고 여기에 둡니다. 이 파일은 `.gitignore`(D-1)에 적어 저장소에 올라가지 않게 막습니다.
로그	프로그램이 남기는 실행 기록입니다. 무엇이 잘못됐는지는 대개 여기에 적혀 있습니다.
캐시	빨리 쓰려고 임시로 저장해 둔 사본입니다. "캐시 지우고 다시 해보세요"는 옛 사본을 버리라는 뜻입니다.
롤백 / 다운타임	롤백은 배포한 것을 이전 버전으로 되돌리는 것, 다운타임은 그동안 서비스가 멈춰 있던 시간입니다.

D-3 개발자와 일할 때 나오는 말

이슈 / 티켓	할 일이나 버그 한 건을 적어 둔 카드입니다. "티켓 끊어 주세요"는 그 카드를 만들어 달라는 말입니다.
스프린트	2주처럼 짧은 주기로 끊어서 일하고, 끝에 결과를 확인하는 방식입니다.
스펙 (spec)	무엇을 어떻게 만들지 미리 정해 둔 약속입니다. 스펙이 없으면 만든 뒤에 말이 달라집니다. 기획 3단의 SPEC(C-1)과 같은 말이고, 개발자와 얘기할 때는 이쪽 이름을 씁니다.
스코프 (범위)	이번에 만들 것과 만들지 않을 것의 경계입니다. 스코프가 늘어난다는 말은 일이 계속 불어난다는 뜻입니다.
MVP	최소한으로 만들어 먼저 내보는 판입니다. 다 만들고 확인하는 대신, 작게 내고 반응을 봅니다.
리팩터링	겉으로 보이는 동작은 그대로 두고 속을 정리하는 일입니다. 기능이 늘지 않아 성과로 보이지 않지만 나중 속도를 좌우합니다.
레거시	오래돼서 손대기 어려워진 기존 코드나 시스템입니다. 나쁘다는 뜻이 아니라 아직 돌아가고 있다는 뜻이기도 합니다.
기술부채	급하게 처리하고 정리를 미뤄 둔 것입니다. 이자처럼 불어나서, 나중에 고치는 비용이 점점 커집니다.
하드코딩	값을 코드 안에 그대로 박아 넣는 것입니다. 당장은 되지만 바꿀 때마다 코드를 고쳐야 합니다. 비밀번호·키를 이렇게 박는 것이 가장 흔한 사고입니다. → D-2 환경변수
사이드이펙트 (부작용)	고친 곳이 아니라 엉뚱한 데가 같이 변하는 것입니다. "이거 고치면 저기 영향 갑니다"가 이 말입니다.

핫픽스	운영 중에 터진 문제를 급히 막는 긴급 수정입니다.
QA	내보내기 전에 품질을 확인하는 단계입니다.
엣지 케이스	드물게 생기는 예외 상황입니다. 평소에는 안 보이다가 실제 데이터에서 터집니다.

D-4 화면에 뜨는 것 읽기

에러 메시지	실패한 이유가 적힌 문장입니다. 전부 읽지 않아도 되고, 그대로 복사해 AI에게 붙여 넣으면 대개 풀립니다.
스택 트레이스	어디서 어떻게 실패했는지 되짚어 나열한 목록입니다. 길지만 맨 위나 맨 아래 한 줄이 핵심입니다.
404 / 500	404는 그런 주소가 없다는 뜻, 500은 서버 쪽이 고장 났다는 뜻입니다. 400번대는 요청 잘못, 500번대는 서버 잘못으로 갈립니다.
JSON	프로그램끼리 주고받는 텍스트 형식입니다. 중괄호와 따옴표로 이름과 값을 짝지어 적습니다.
키 / 값	이름표와 그 내용의 짝입니다. `"이름": "홍길동"`에서 앞이 키, 뒤가 값입니다.
API 키 · 인증 토큰	프로그램이 서비스에 드나들 때 쓰는 출입증입니다. 남에게 보이면 안 됩니다. 글자를 세는 단위인 토큰(A-3)과는 다른 말입니다.
타임아웃	정해진 시간 안에 응답이 오지 않아 포기한 상태입니다. 고장이 아니라 늦어서 끊긴 경우가 많습니다.
레이트 리밋	정해진 시간 동안 쓸 수 있는 횟수·분량(토큰) 제한입니다. 걸리면 잠시 기다렸다 다시 하면 됩니다.

E. Claude Code 곁에 두고 보기

E-1 키보드 단축키

Enter	메시지를 보냅니다. 요청을 다 썼을 때 누릅니다.
\ + Enter	줄을 바꿉니다. 백슬래시를 치고 Enter를 누르면 됩니다. 설정이나 키 충돌 없이 어디서나 됩니다.
Esc	진행 중인 작업을 멈춥니다. 잘못 시켰거나 방향을 바꿀 때 누릅니다.
Esc 두 번	이전 지점으로 되돌립니다. 앞선 메시지로 돌아가 다시 시킬 때 씁니다.
↑ (위 화살표)	직전에 보낸 입력을 다시 꺼냅니다. 비슷한 요청을 조금만 고쳐 다시 보낼 때 편합니다.
Ctrl + L	화면을 정리해 다시 그립니다. 글씨가 깨져 보일 때 씁니다.

Ctrl + D	종료합니다. `/exit`와 같습니다. Ctrl + C를 두 번 눌러도 같습니다. 참고로 터미널에서 Ctrl + C는 복사가 아니라 중단입니다.
Shift + Tab	권한 모드를 바꿉니다. 자동 승인이 편해 보이지만 처음에는 기본 모드를 권합니다. 계획을 직접 읽고 승인하는 습관이 먼저입니다.

E-2 입력 앞에 붙이는 기호

`/` (슬래시)	명령 메뉴를 엽니다. 치기 시작하면 목록이 떠서 고르면 됩니다.
`@` (골뱅이)	파일을 콕 집어 지정합니다. 예 - `@월별매출.csv 읽고 정리해줘`

E-3 자주 쓰는 슬래시 명령

`/clear`	새 대화를 시작합니다(맥락 비우기). 주제가 바뀔 때 씁니다.
`/help`	쓸 수 있는 명령 전체 목록을 띄웁니다. 도구가 업데이트되면 이 표보다 `/help`가 최신입니다.
`/resume`	이전 대화를 다시 엽니다.
`/compact`	긴 대화를 요약해 기억할 공간을 확보합니다.
`/exit`	도구를 끝냅니다. 다른 폴더에서 다시 시작할 때 씁니다.

E-4 손에 익힐 것과 함정

외울 것은 셋	Enter(보내기) · Esc(멈추기) · `/clear`(새 대화). 이 셋만 손에 익히고 나머지는 이 표를 꺼내 쓰면 됩니다.
긴 지시는 메모장에서	터미널에서는 줄 중간에 Enter가 눌리면 미완성 문장이 그대로 전송됩니다. 긴 지시는 메모장에 써서 통째로 붙여넣으세요. 붙여넣기로 들어간 줄바꿈은 전송되지 않습니다.

F. 그대로 복사해 쓰는 프롬프트

F-1 확인할 때

지어내지 않게 막기

내부 자료를 정리시킬 때, 지시문 맨 아래에 붙입니다

```
# 제약
추측하거나 매끄럽게 채워 넣지 마. 원본에 '?'나 '확인 필요'가 달린
항목은 확정 사실로 적지 말고 '미확인'으로 남겨줘. 자료에 없는 내용은
만들지 마.
```

"자료에 없음"을 말하게 하기

빈칸을 지어낼 것 같을 때. 이 한 줄이 환각을 크게 줄입니다

```
자료에 있는 값만 사용해. 자료에 없으면 추측하지 말고
"자료에 없음"이라고 그대로 표기해.
```

근거 캐묻기

결과의 숫자나 문장이 미심쩍을 때, 그 자리에서 따집니다

```
이 숫자는 원본의 어느 줄을 근거로 했는지 밝혀줘.
```

F-2 막혔을 때

에러 그대로 넘기기

빨간 글씨가 떴을 때. 에러는 실패가 아니라 다음 지시의 재료입니다

```
방금 이런 에러가 났어. 무슨 뜻이고 어떻게 해결하면 되는지 알려주고,
고쳐서 다시 실행해줘.

(여기에 에러 메시지를 그대로 붙여넣기)
```

환경 문제도 말로 넘기기

파일이 열려 있어 막히는 등, 도구 바깥 사정으로 멈췄을 때

```
파일이 열려 있어서 막혔으면, 닫은 뒤 다시 시도해줘.
```

이어서 콕 집어 고치기

결과가 조금 어긋났을 때. 처음부터 다시 시키지 않습니다

```
좋아. 그런데 '다음 회의 전 확인할 항목'을 우선순위 순으로 정렬하고,
각 항목 옆에 '왜 중요한지'를 한 줄씩 붙여줘.
```

F-3 시킬 때

지시문 빈칸 틀 (프롬프트 5원칙)

무엇부터 써야 할지 막막할 때. 맨 위에 할 일 한 줄을 쓰고 다섯 칸을 채우면 지시문이 됩니다

```
(  )를 (  )해줘.

# 역할
너는 (  )이다.

# 맥락
아래 자료는 (  )이다. (확인 전인 항목이 있으면 여기에 적는다)

# 제약
추측하지 마. 자료에 없는 내용은 만들지 마.
확인 필요 표시는 그대로 남겨줘.

# 형식
(  ) 모양으로, 각 항목 (  ) 이내로 정리해줘.

# 예시 (형식만으로 부족할 때만)
(  )
```

형식과 견본 지정

원하는 모양이 말로는 잘 전달되지 않을 때

```
# 형식
각 지표를 한 줄씩, '지표명: 값 (확인 / 미확인)' 형태로 정리해줘.

# 예시
고객 수: 140개사 (미확인)
월 이탈률: 1.5% (확인)

아래 예시는 형식 견본일 뿐, 내용은 자료에서 가져와.
```

칸을 정해 매번 같은 자리에 받기

같은 일을 반복해서 시킬 때. 결과가 들쭉날쭉하지 않게 만듭니다

> 아래 양식으로 출력해줘.
> <by_category>카테고리별 합계 (표)</by_category>
> <total>총 합계 (재계산해 검증)</total>
> <over_budget>한도 초과 항목·사유 확인 요청 (없으면 "없음")</over_budget>
> <note>그 외 특이사항</note>

요구사항 메모 3줄

만들거나 고치기 전에. 이 3줄이 곧 채점 기준이 됩니다

> [요구사항 메모 - (작업 이름)]
> 1. 무엇을: ( )
> 2. 어떤 형식으로: ( )
> 3. 하지 말 것: 자료에 없는 내용·숫자를 지어내지 않는다.
> 확실하지 않으면 [확인 필요]로 표기한다.

F-4 고쳐 달라고 할 때

요약이 너무 길거나 짧을 때

분량이 안 맞을 때. 숫자로 못 박습니다

> 요약은 정확히 3줄로, 각 줄 40자 이내로 해줘.

표로 받고 싶을 때

줄글로 왔는데 표가 필요할 때

> 다음 주 계획은 표로(항목·내용·비고 열) 만들어줘.

말투가 안 맞을 때

너무 딱딱하거나 가벼울 때

> 사내 보고서 톤(간결한 존댓말)으로 다시 써줘.

지어낸 것 같을 때

원본에 없던 내용이 보일 때

> 메모에 없는 건 지어내지 말고 [확인 필요]로 표시해줘.

숫자가 틀렸을 때

값이 원본과 다를 때

> 매출 같은 숫자는 메모(원본) 그대로 옮겨줘.

G. 막혔을 때

G-1 도구가 안 될 때

답이 갑자기 엉뚱해졌다	대화가 길어져 책상(컨텍스트)이 넘친 것입니다. **어느 도구에서나** - 요점 세 줄만 정리해 새 대화로 옮겨 시작하세요. **Claude Code라면** - `/compact`로 요약해 공간을 확보하거나 `/clear`로 새로 시작합니다.
앞에서 한 얘기를 잊은 것 같다	같은 원인입니다. **어느 도구에서나** - 중요한 전제는 다시 한 번 적어 주세요. **저장 기능이 있는 도구라면** - 규칙 파일(CLAUDE.md)이나 프로젝트 지시문에 넣어 매번 적용되게 합니다.
빨간 글씨(에러)가 떴다	통째로 복사해 그대로 붙여넣고 맡깁니다. 에러는 실패가 아니라 다음 지시의 재료입니다. 어느 도구에서나 같습니다. → F-2
파일을 못 찾는다고 한다	**어느 도구에서나** - 그 도구가 볼 수 있는 자리에 파일이 있는지 먼저 확인하세요(챗 도구는 첨부, Cowork는 연결된 폴더). **Claude Code라면** - 실행한 폴더 안만 봅니다. `/exit`로 나가 올바른 폴더로 이동한 뒤 다시 실행하세요.
만든 규칙이나 서브에이전트가 안 먹는다	**어느 도구에서나** - 저장이 됐는지, 지금 대화에 적용되는 자리에 넣었는지 확인하세요. **Claude Code라면** - 파일로 만든 것은 다시 시작해야 인식됩니다(`/clear`로는 안 됩니다). 여러 개를 만들었다면 다 만든 뒤 한 번만 재시작하세요.
권한을 허락하겠냐는 창이 떴다	무엇을 허락하는지 한 줄 읽고 누릅니다. 모르겠으면 일단 거절하고 "방금 뭘 하려던 거야?"라고 물어보면 됩니다. 어느 도구에서나 같습니다.
화면 글씨가 깨져 보인다	**브라우저 도구라면** 새로고침(F5). **Claude Code라면** Ctrl + L로 화면을 다시 그립니다.
너무 느리거나 멈춘 것 같다	**어느 도구에서나** - 범위를 좁혀 다시 시킵니다. 한 번에 다 시키는 것보다 쪼개는 쪽이 대체로 빠릅니다. **Claude Code라면** - Esc로 멈추고 다시 시킵니다.

G-2 결과가 미덥지 않을 때

숫자가 수상하다	세 단계로 봅니다 - 원본과 한 줄씩 대조 → 합계·비율을 직접 다시 계산 → "자료에 없으면 없다고 말해"를 걸고 재실행. → F-1
원본에 없던 내용이 보인다	환각입니다. 제약을 걸고 다시 시키세요. 정갈하게 정리될수록 더 믿고 싶어지므로, 그럴 때일수록 원본을 펼쳐 둡니다. → F-1

출처를 달았는데 그 내용이 아니다	링크를 눌러 두 가지를 따로 확인합니다 - 그 페이지가 실제로 있는가, 그 안에 그 내용이 있는가. 출처가 달렸다는 것과 맞다는 것은 별개입니다.
결과가 조금 어긋난다	처음부터 다시 시키지 말고, 같은 대화에서 어긋난 곳만 콕 집어 고쳐 달라고 하세요. → F-2
잘 되던 게 고친 뒤 이상해졌다	회귀입니다. 예전에 잘 되던 입력을 다시 넣어 어디서부터 어긋나는지 확인하세요. 고칠 때마다 같은 입력으로 재확인하는 습관이 이걸 막습니다.

G-3 파일·저장에서 막힐 때

결과가 파일로 저장되지 않는다	폴더와 파일명을 콕 집어 다시 시켜 보세요 - "연결된 (폴더 이름)에 (파일 이름)으로 저장해줘". 저장이 안 된다는 설명이 돌아와도 이 문구로 다시 시키면 되는 경우가 많습니다.
파일이 열려 있어 막힌다	도구 바깥 사정도 말로 넘기면 됩니다 - "파일이 열려 있어서 막혔으면, 닫은 뒤 다시 시도해줘". → F-2
원본이 사라질까 걱정된다	먼저 사본을 만들게 하고 사본에서 작업시키세요. 기록이 쌓이는 작업은 덮어쓰기 대신 이어붙이기로 시킵니다.

H. 더 넓은 IT 용어 (지금 몰라도 됩니다)

H-1 클라우드와 서버

클라우드 (AWS·Azure·GCP)	남의 컴퓨터를 필요한 만큼 빌려 쓰는 것입니다. 사무실에 서버를 두는 대신 인터넷 너머의 컴퓨터를 씁니다. AWS·Azure·GCP가 대표적입니다.
온프레미스	회사 안에 서버를 직접 두고 쓰는 방식입니다. 클라우드의 반대말이고, 보안 규정 때문에 이렇게만 하는 회사도 많습니다.
SaaS	설치 없이 웹으로 바로 쓰고 구독료를 내는 소프트웨어입니다. Google Sheets·Notion이 그 예입니다.
서버리스	서버를 직접 관리하지 않고 코드만 올려 두면 필요할 때만 돌아가는 방식입니다. 서버가 없다는 뜻이 아니라 신경 쓸 일이 없다는 뜻입니다.
컨테이너 · 도커	프로그램을 실행에 필요한 것까지 통째로 담아 옮기는 상자입니다. "제 컴퓨터에선 되는데요"를 줄이려고 씁니다. → D-2 로컬/서버
트래픽	서비스에 들어오는 접속량입니다. 몰리면 느려지거나 멈춥니다.
스케일링 (증설)	접속이 늘 때 처리 능력을 늘리는 것입니다. 계산대를 더 여는 것에 가깝습니다.
IP 주소 · 포트	IP는 인터넷상의 집 주소, 포트는 그 건물의 몇 호실인지에 해당합니다.

백업 · 이중화	백업은 사본을 따로 보관하는 것, 이중화는 하나가 고장 나도 다른 하나가 대신하도록 둘을 두는 것입니다.
가용성 · SLA	가용성은 서비스가 멈추지 않고 돌아가는 정도, SLA는 "얼마나 안 멈추겠다"고 약속한 문서입니다.
마이그레이션	데이터나 시스템을 다른 곳으로 옮기는 일입니다. 이사에 해당하고, 대개 예상보다 오래 걸립니다.

H-2 회사에서 듣는 보안 용어

이 묶음은 회사 정보보안 쪽 말입니다. AI에 자료를 올려도 되는지 판단하는 쪽은 → C-2 입니다.

2단계 인증 (2FA·OTP)	비밀번호에 더해 휴대폰 숫자 같은 두 번째 확인을 거치는 것입니다. 비밀번호가 새도 한 겹이 더 남습니다.
SSO (통합 로그인)	한 번 로그인하면 회사 여러 서비스를 다시 로그인하지 않고 쓰는 방식입니다.
VPN	바깥에서 회사 내부망에 안전하게 접속하는 통로입니다. 재택근무 때 켜라고 하는 그것입니다.
방화벽	오가는 통신 중 허락된 것만 통과시키는 문지기입니다.
암호화	내용을 열쇠 없이는 못 읽게 바꾸는 것입니다. 오갈 때(전송 중)와 저장돼 있을 때 각각 적용합니다.
피싱 · 스미싱	진짜인 척 속여 비밀번호나 결제 정보를 빼내는 수법입니다. 피싱은 주로 메일, 스미싱은 문자입니다.
랜섬웨어	파일을 암호로 잠그고 돈을 요구하는 악성 프로그램입니다. 백업이 유일한 확실한 대비책입니다.
악성코드 (멀웨어)	해를 끼치려고 만든 프로그램을 통틀어 부르는 말입니다.
취약점 · 패치	취약점은 공격에 뚫릴 수 있는 약한 곳, 패치는 그걸 막는 수정입니다. "업데이트하세요"는 대개 이 얘기입니다.
망분리	업무망과 인터넷망을 물리적·논리적으로 갈라 두는 것입니다. 이 환경에서는 외부 AI 도구 접속이 막혀 있을 수 있습니다.
접근 권한 · 계정 분리	누가 무엇까지 볼 수 있는지 정해 두고, 역할이 다르면 계정도 나누는 것입니다. → C-2 최소 권한
감사 로그	누가 언제 무엇을 했는지 남기는 기록입니다. 사고가 났을 때 되짚는 근거가 됩니다.

H-3 AI가 짜 준 코드 읽기

파이썬 (Python)	배우기 쉬워 자동화·데이터 처리에 가장 많이 쓰이는 프로그래밍 언어입니다. AI가 짜 주는 코드도 대개 이 언어입니다.
라이브러리 · 패키지	남이 만들어 둔 기능 묶음입니다. 매번 새로 만들지 않고 가져다 씁니다.
import (가져오기)	코드 맨 위에서 "이 라이브러리를 쓰겠다"고 선언하는 줄입니다.

pip · 설치	라이브러리를 내려받아 설치하는 명령입니다. `pip install ...` 형태로 나옵니다. 없다는 에러가 나면 대개 이걸 안 한 것입니다.
가상환경	프로젝트마다 라이브러리를 따로 담아 두는 칸막이입니다. 서로 버전이 부딪히지 않게 합니다.
변수	값에 이름을 붙여 담아 두는 상자입니다.
반복문 · 조건문	반복문은 같은 일을 여러 번, 조건문은 "~이면 이렇게"입니다. 자동화 코드의 대부분이 이 둘의 조합입니다.
주석	`#`으로 시작하는, 실행되지 않는 설명 줄입니다. AI에게 "여기를 고쳐 달라"고 표시할 때도 씁니다.
들여쓰기	줄 앞의 공백입니다. 파이썬은 이걸로 코드의 묶음을 판단해서, 어긋나면 실행이 안 됩니다.

H-4 이름만 알아두면 되는 것들

pandas · openpyxl	표와 엑셀을 다루는 파이썬 라이브러리입니다. AI가 엑셀 작업 코드를 짜면 대개 이 이름이 나옵니다.
requests · BeautifulSoup	웹에서 자료를 가져오고(requests) 필요한 부분만 골라내는(BeautifulSoup) 라이브러리입니다.
matplotlib	차트를 그리는 라이브러리입니다.
Flask · FastAPI	간단한 웹 서버를 만드는 파이썬 도구입니다.
React · Next.js	웹 화면을 만드는 도구입니다. 바이브 코딩으로 만든 웹앱이 대개 이 형태입니다.
Node.js	브라우저 밖에서 자바스크립트를 돌리는 실행 환경입니다.
Jupyter · Colab	코드와 결과를 한 화면에서 보며 돌리는 노트 형태의 도구입니다. Colab은 설치 없이 브라우저에서 씁니다.
Streamlit	파이썬 몇 줄로 화면이 있는 도구를 만드는 라이브러리입니다.
오픈소스	소스 코드를 공개해 누구나 쓰고 고칠 수 있게 한 소프트웨어입니다. 위 이름들이 대부분 여기에 해당합니다.
이름은 바뀝니다	이 묶음의 도구 이름은 유행을 탑니다(기준 2026-07). 이름을 외우기보다 "표를 다루는 것"·"화면을 만드는 것"처럼 역할로 기억하세요. 궁금하면 AI에게 "무슨 라이브러리를 썼고 왜 골랐는지 한 줄로 설명해줘"라고 물으면 됩니다.

H-5 공장·설비에서 듣는 말

펌웨어	기계 안에 심어 둔 소프트웨어입니다. 설비·센서 같은 장비가 어떻게 움직일지를 정하고, 업데이트하면 장비 동작이 바뀝니다.

임베디드	기계 안에 들어가 그 기계만을 위해 도는 컴퓨터입니다. 자동차·설비·가전 안에 들어 있습니다.
PLC	공장 설비를 순서대로 제어하는 산업용 제어기입니다. 컨베이어·밸브 같은 것을 움직입니다.
센서 · 액추에이터	센서는 온도·압력처럼 상태를 재는 장치, 액추에이터는 실제로 움직이는 장치입니다.
IoT · 엣지	IoT는 설비·기기를 인터넷에 연결해 데이터를 모으는 것, 엣지는 그 데이터를 중앙으로 보내기 전에 현장에서 먼저 처리하는 방식입니다.
SCADA	설비 상태를 한 화면에서 감시하고 제어하는 시스템입니다. 공장의 상황판에 해당합니다.
MES · ERP	MES는 생산 현장의 실행을 관리하는 시스템, ERP는 회사 전체의 자원·회계를 관리하는 시스템입니다.
디지털 트윈	실제 설비를 컴퓨터 안에 똑같이 만들어 두고 미리 돌려 보는 것입니다.
예지보전	고장 나기 전에 데이터로 낌새를 잡아 미리 손보는 방식입니다. 고장 난 뒤에 고치는 사후보전의 반대입니다.

찾아보기

오른쪽은 그 단어가 실린 묶음과 쪽수입니다. 예를 들어 **A-3 · 5**는 A장 세 번째 묶음(AI 기본), 5쪽입니다.

한글

가드레일	C-3 · 9	깃	D-1 · 11	로컬	D-2 · 11
가명	C-2 · 9	깃허브	D-1 · 11	로컬 스크립트	B-1 · 6
가상환경	H-3 · 20	깃헙	D-1 · 11	롤백	D-2 · 11
가용성	H-1 · 19	끄기	E-1 · 13	리팩터링	D-3 · 12
가져오기	H-3 · 20	나가기	E-3 · 14	리포	D-1 · 11
감사 기록	H-2 · 20	노드	H-4 · 21	마스킹	C-2 · 9
감사 로그	H-2 · 20	느림	G-1 · 18	마이그레이션	H-1 · 19
값	D-4 · 13	다운타임	D-2 · 11	마크다운	A-2 · 5
개선 로그	C-3 · 9	단계	C-1 · 8	말투가 안 맞을 때	F-4 · 17
개인정보	C-2 · 9	단축키	E-1 · 13	망분리	H-2 · 20
검산	C-1 · 8	대시보드	C-4 · 10	맥락	A-3 · 5
검색 증강	A-3 · 5	대화 되돌리기	E-1 · 13	머지	D-1 · 11
검증	C-3 · 9	더미 데이터	C-2 · 9	멀웨어	H-2 · 20
경로	A-2 · 5	덮어쓰기	A-2 · 5	멀티모달	A-3 · 5
계정	B-3 · 7	데이터 4계층	C-2 · 9	멀티스텝	B-1 · 6
계정 분리	H-2 · 20	데이터베이스	D-2 · 11	멈춤	G-1 · 18
고친 뒤 이상해짐	G-2 · 18	도구 호출	B-1 · 6	메인 에이전트	B-1 · 6
골든셋	C-3 · 9	도메인	D-2 · 11	메타데이터	C-4 · 10
골뱅이	E-2 · 14	도커	H-1 · 19	명령 목록	E-3 · 14
구독형	H-1 · 19	되돌리기	D-1 · 11	명령어	B-2 · 7
권한	B-3 · 7	들여쓰기	H-3 · 20	명령어	E-3 · 14
권한 모드	E-1 · 13	디지털 트윈	H-5 · 21	명령창	A-1 · 5
권한 팝업	G-1 · 18	라이브러리	H-3 · 20	모델	A-3 · 5
규칙서	B-2 · 7	랜섬웨어	H-2 · 20	목표선	C-4 · 10
규칙이 안 먹음	G-1 · 18	레거시	D-3 · 12	문제 정의 카드	C-1 · 8
근거 캐묻기	F-1 · 15	레이트 리밋	D-4 · 13	바이브 코딩	B-1 · 6
기반 모델	A-3 · 5	레포	D-1 · 11	반복문	H-3 · 20
기술부채	D-3 · 12	레포지토리	D-1 · 11	방화벽	H-2 · 20
기준 문서	C-1 · 8	로그	D-2 · 11	배포	D-2 · 11
기획 3단	C-1 · 8	로그인	B-3 · 7	백업	H-1 · 19

백엔드 D-2 · 11
범위 D-3 · 12
벤더 정책 C-2 · 9
벤치마크 C-3 · 9
변수 H-3 · 20
병렬 B-1 · 6
보안 등급 C-2 · 9
본선 D-1 · 11
부작용 D-3 · 12
붙여넣기 E-4 · 14
브라우저 A-1 · 5
브라우저 확장 B-3 · 7
브랜치 D-1 · 11
빌드 D-2 · 11
빨간 글씨 D-4 · 13
빨간 글씨 G-1 · 18
사물인터넷 H-5 · 21
사용량 제한 D-4 · 13
사이드이펙트 D-3 · 12
사후보전 H-5 · 21
상태 코드 D-4 · 13
생성형 AI A-3 · 5
서버 D-2 · 11
서버리스 H-1 · 19
서브에이전트 B-1 · 6
서브에이전트 G-1 · 18
설치 H-3 · 20
성공 지표 C-1 · 8
센서 H-5 · 21
순차 B-1 · 6
숫자가 이상함 G-2 · 18
숫자가 틀렸을 때 F-4 · 17
스미싱 H-2 · 20
스케일링 H-1 · 19
스코프 D-3 · 12
스크립트 B-4 · 8
스택 트레이스 D-4 · 13
스테이징 D-2 · 11
스펙 D-3 · 12
스프레드시트 B-4 · 8
스프린트 D-3 · 12
슬래시 E-2 · 14
슬래시 명령 B-2 · 7
승인 B-3 · 7
승인 창 G-1 · 18
시간 트리거 B-4 · 8
시트 ID B-4 · 8
악성코드 H-2 · 20
암호화 H-2 · 20
압축 A-2 · 5
앞 내용을 잊음 G-1 · 18
앞 메시지로 E-1 · 13
액추에이터 H-5 · 21
업데이트 H-2 · 20
업로드 C-2 · 9
엉뚱한 답 G-1 · 18
에러 D-4 · 13
에러 그대로 넘기기 F-2 · 15
에러 메시지 D-4 · 13
에러가 떴을 때 G-1 · 18
에이전트 B-1 · 6
엑셀 A-2 · 5
엣지 H-5 · 21
엣지 케이스 D-3 · 12
역할 A-3 · 5
예시 제시 A-3 · 5
예약 실행 B-4 · 8
예지보전 H-5 · 21
오류 D-4 · 13
오류 G-1 · 18
오픈소스 H-4 · 21
온프레미스 H-1 · 19
올려도 되나 C-2 · 9
외울 것은 셋 E-4 · 14
요구사항 메모 3줄 F-3 · 16
요약이 너무 길거나 짧을 때 F-4 · 17
운영 D-2 · 11
워크스페이스 B-3 · 7
원본 보존 G-3 · 19
위 화살표 E-1 · 13
유행 H-4 · 21
이름은 바뀝니다 H-4 · 21
이상해짐 G-1 · 18
이슈 D-3 · 12
이어붙이기 A-2 · 5
이어서 콕 집어 고치기 F-2 · 15
이전 H-1 · 19
이중화 H-1 · 19
인덴트 H-3 · 20
인증 B-3 · 7
인증 토큰 D-4 · 13
임베디드 H-5 · 21
자산 C-4 · 10
자체 서버 H-1 · 19
작업 폴더 A-2 · 5
재시작 G-1 · 18
저장소 D-1 · 11
저장이 안 됨 G-3 · 19
전송 E-1 · 13
접근 권한 H-2 · 20
접속량 H-1 · 19
제미나이 A-3 · 5
제약 A-3 · 5
제어기 H-5 · 21
조건문 H-3 · 20
조금 어긋남 G-2 · 18
종료 E-1 · 13
종료 E-3 · 14

주석 H-3 · 20
줄바꿈 E-1 · 13
증설 H-1 · 19
지시문 A-3 · 5
지시문 빈칸 틀 (프롬프트 5원칙) F-3 · 16
지어내지 않게 막기 F-1 · 15
지어낸 것 같을 때 F-4 · 17
지어낸 내용 G-2 · 18
지어냄 A-3 · 5
지피티 A-3 · 5
차트 H-4 · 21
채점 기준표 C-3 · 9
챗봇 B-1 · 6
챗GPT A-3 · 5
첨부 C-2 · 9
최소 권한 C-2 · 9
최소 단축키 E-4 · 14
출처가 틀림 G-2 · 18
충돌 D-1 · 11
취약점 H-2 · 20
칸을 정해 매번 같은 자리에 받기 F-3 · 16
캐시 D-2 · 11
커넥터 B-3 · 7
커밋 D-1 · 11
커밋 취소 D-1 · 11
컨테이너 H-1 · 19
컨텍스트 윈도우 A-3 · 5
코랩 H-4 · 21
크롤링 H-4 · 21
클라우드 H-1 · 19
클로드 A-3 · 5
클론 D-1 · 11
키 D-4 · 13
타임아웃 D-4 · 13
터미널 A-1 · 5
템플릿 C-4 · 10
토큰 A-3 · 5
통합 로그인 H-2 · 20
트래픽 H-1 · 19
트리거 B-4 · 8
티켓 D-3 · 12
파이썬 H-3 · 20
파이프라인 B-1 · 6
파일 A-2 · 5
파일 올리기 C-2 · 9
파일 지정 E-2 · 14
파일럿 C-1 · 8
파일을 못 찾음 G-1 · 18
파일이 열려 있음 G-3 · 19
패치 H-2 · 20
패키지 H-3 · 20
펌웨어 H-5 · 21
페인포인트 C-1 · 8
편집기 A-1 · 5
포트 H-1 · 19
폴더 A-2 · 5
표로 받고 싶을 때 F-4 · 17
푸시 D-1 · 11
풀 D-1 · 11
풀 리퀘스트 D-1 · 11
프로덕션 D-2 · 11
프론트엔드 D-2 · 11
프롬프트 A-3 · 5
프롬프트 5원칙 A-3 · 5
플러그인 B-2 · 7
피싱 H-2 · 20
하네스 B-2 · 7
하드코딩 D-3 · 12
할루시네이션 A-3 · 5
함수 B-4 · 8
핫픽스 D-3 · 12
협업 B-1 · 6
형식과 견본 지정 F-3 · 16
호스팅 D-2 · 11
화면 깨짐 G-1 · 18
확산 계획서 C-1 · 8
확장자 A-2 · 5
환각 A-3 · 5
환경 문제도 말로 넘기기 F-2 · 15
환경변수 D-2 · 11
회귀 확인 C-3 · 9

영문 · 기호

.csv A-2 · 5
.env D-2 · 11
.gitignore D-1 · 11
.html A-2 · 5
.md A-2 · 5
.xlsx A-2 · 5
.zip A-2 · 5
"자료에 없음"을 말하게 하기 F-1 · 15
/clear E-3 · 14
/compact E-3 · 14
/exit E-3 · 14
/help E-3 · 14
/resume E-3 · 14
2FA H-2 · 20
2단계 인증 H-2 · 20
3단 스케치 B-1 · 6
3층위 C-3 · 9
5 Whys C-1 · 8

5원칙 A-3 · 5
403 D-4 · 13
404 D-4 · 13
429 D-4 · 13
500 D-4 · 13
Action B-1 · 6
Agent B-1 · 6
AI Validation C-3 · 9
API B-3 · 7
API 키 D-4 · 13
append A-2 · 5
Apps Script B-4 · 8
As-is C-1 · 8
AWS H-1 · 19
Azure H-1 · 19
BeautifulSoup H-4 · 21
Benchmark C-3 · 9
branch D-1 · 11
build D-2 · 11
cache D-2 · 11
Chatbot B-1 · 6
Claude Code B-2 · 7
CLAUDE.md B-2 · 7
CLI B-2 · 7
clone D-1 · 11
CMD A-1 · 5
Colab H-4 · 21
comment H-3 · 20
commit D-1 · 11
conflict D-1 · 11
Cowork Project B-2 · 7
Ctrl+C E-1 · 13
Ctrl+D E-1 · 13
Ctrl+L E-1 · 13
dashboard C-4 · 10
DB D-2 · 11
deploy D-2 · 11
Docker H-1 · 19
E2E B-1 · 6
embedded H-5 · 21
encryption H-2 · 20
End-to-End B-1 · 6
Enter E-1 · 13
ERP H-5 · 21
ERRC C-1 · 8
Esc E-1 · 13
Esc 두 번 E-1 · 13
FastAPI H-4 · 21
Few-shot A-3 · 5
firewall H-2 · 20
firmware H-5 · 21
Flask H-4 · 21
for문 H-3 · 20
Foundation Model A-3 · 5
Function Calling B-1 · 6
GCP H-1 · 19
Git D-1 · 11
GitHub D-1 · 11
Golden Set C-3 · 9
Google Sheets B-4 · 8
Guardrail C-3 · 9
Hallucination A-3 · 5
hosting D-2 · 11
Human Eval C-3 · 9
if문 H-3 · 20
import H-3 · 20
Inter-rater Agreement C-3 · 9
IoT H-5 · 21
IP H-1 · 19
IP 주소 H-1 · 19
Job to be Done C-1 · 8
JSON D-4 · 13
Jupyter H-4 · 21
KPI C-4 · 10
library H-3 · 20
Live Artifacts B-2 · 7
LLM A-3 · 5
LLM-as-a-Judge C-3 · 9
log D-2 · 11
Long Bet C-1 · 8
LV1 C-2 · 9
LV2 C-2 · 9
LV3 C-2 · 9
LV4 C-2 · 9
main D-1 · 11
master D-1 · 11
matplotlib H-4 · 21
MCP B-3 · 7
merge D-1 · 11
MES H-5 · 21
Model C-3 · 9
MVP D-3 · 12
Next.js H-4 · 21
Node.js H-4 · 21
open source H-4 · 21
openpyxl H-4 · 21
OTP H-2 · 20
Output B-1 · 6
pandas H-4 · 21
PBL C-1 · 8
Phase C-1 · 8
pip H-3 · 20
PLAN C-1 · 8
PLC H-5 · 21
PowerShell A-1 · 5
PR D-1 · 11
PRD C-1 · 8
PRD 메모 C-1 · 8

Product C-3 · 9
pull D-1 · 11
Pull Request D-1 · 11
push D-1 · 11
Python H-3 · 20
QA D-3 · 12
Quick Win C-1 · 8
RAG A-3 · 5
React H-4 · 21
README D-1 · 11
Regression C-3 · 9
repo D-1 · 11
requests H-4 · 21
revert D-1 · 11
Rubric C-3 · 9
SaaS H-1 · 19
SCADA H-5 · 21
SCAMPER C-1 · 8
serverless H-1 · 19
Shift+Tab E-1 · 13
Skill B-2 · 7
skill-creator B-2 · 7
SKILL.md B-2 · 7
SLA H-1 · 19
SMART 목표 C-1 · 8
SPEC C-1 · 8
spec D-3 · 12
SSO H-2 · 20
Streamlit H-4 · 21
Sub-goal B-1 · 6
System C-3 · 9
Test Case C-3 · 9
To-be C-1 · 8
Tool Use B-1 · 6
Trigger B-1 · 6
URL D-2 · 11
venv H-3 · 20
VPN H-2 · 20
VS Code A-1 · 5
XML 태그 A-3 · 5
ZDR C-2 · 9

FlowCoder

강의가 끝나도 계속 쓰세요

이 사전은 오늘 강의 수강생을 위해 만든 자료입니다.
가져가서 쓰시고, 필요하면 팀에 공유하셔도 됩니다.

강의 내용 질문

오늘 다룬 내용 중 막히는 부분이 있으면 메일로 물어보셔도 됩니다.

admin@flow-coder.com

검색해서 찾는 웹판 - https://flowcoder-ai-glossary.vercel.app

플로우코더(FlowCoder) - AI가 일하게 만드는 회사

플로우코더(FlowCoder) | 사업자등록번호 374-16-02889
대표자 조용현, 박현일 | 경기도 남양주시 홍유릉로248번길 26, 지하1층(금곡동)